

# ENTROPY REDUCTION VIA SIMPLIFIED IMAGE CONTOURIZATION

Martin J. Turner  
University of Cambridge Computer Laboratory  
New Museums Site, Pembroke Street  
Cambridge. CB2 3QG. ENGLAND

**Abstract.** The process of *contourization* is presented which converts a raster image into a set of plateaux or contours. These contours can be grouped into a hierarchical structure, defining total spatial inclusion, called a *contour tree*. A *contour coder* has been developed which fully describes these contours in a compact and efficient manner and is the basis for an image compression method. Simplification of the contour tree has been undertaken by merging contour tree nodes thus lowering the contour tree's entropy. This can be exploited by the contour coder to increase the image compression ratio. By applying general and simple rules derived from physiological experiments on the human vision system, lossy image compression can be achieved which minimises noticeable artifacts in the simplified image.

## 1. Contourization -- The Basic Principles

*Contourization* is the creation of an image as a contoured landscape with a one-to-one or one-to-many mapping from pixel values to pseudo height values. An image can be fully described and represented as a set of contours where each contour is a region whose perimeter pixels' values have similar magnitude. Contours can have within their boundaries other contours but no boundary edge pixel is allowed to be part of more than one contour boundary.

A contour description of an image offers two advantageous features over a standard raster image format:

- Image operations can be carried out on contours in one step rather than at the pixel level, giving certain image manipulations potential speedups.
- This representation potentially allows a more compact description format for reduced storage and transmission time.

The rest of this paper is divided into two main sections, firstly a coding method to represent a contour description as compactly as possible and then a new method for lossy compression by exploiting this representation.

## 2. Contour Description and Coding

A contour description of an image is not a new idea and many related edge description techniques have been studied. These descriptions have all been used, in the past, as aids to image coding. Presented is an up to date implementation which performs as well if not better than most of the current image representations.

## 2.1 Contour Description

To describe a contour fully three distinct pieces of information are sufficient:<sup>1</sup>

**Start Location** of the contour in the x-y space of the raster image. This is defined as the top left hand corner of the contour, as in a raster scan of the original image.

**Height Value** of the contour's perimeter.

**Boundary Description** of the contour. This is a representation of the shape of its perimeter.

The process of extracting contour information from images can be achieved in many ways which divide into two broad methods, either linear with the scan remembering a current list of half seen contours or processing contours as they are first seen, in a raster scan. The latter method was developed for coding efficiency reasons and a contour following technique commonly called a Backtracking Bug Follower was used as described in [8].

## 2.2 Contour Coding

Once the contours have been isolated they can be coded producing the three streams of information as stated above.

Preceding these streams of information is a very small header containing three items:

- *Horizontal Size* of the original raster image in pixels.
- *Vertical Size* of the original raster image in pixels.
- *Magic Number* specifying which version coded this image. This is purely for development purposes so versions can be compared.

As an aid towards efficient coding, contours are coded in the order they are first seen in a raster scan sense. This means that when the decoder is dealing with the  $(n + 1)$ th contour it knows all about the previous  $n$  contours and thus all about every pixel's value before the  $(n + 1)$ th contours' start location. This ordering has the advantage over the obvious alternative of ordering by height as it aids both the coding of start locations and boundary descriptions.

The next three parts deal with coding the three pieces of information for the  $(n + 1)$ th contour.

**2.2.1 Start Locations.** The start location is coded as the next possible offset from the last contours' start location, reduced by the number of impossible start location known by the decoder at the time of decoding. Impossible start locations known by the decoder consist of the set of boundary points of the previous  $n$  contours. Figure 1 shows a case where contours W and X have been coded and any pixel after contour X's start location not on a known boundary is a potential start location for the next contour.

Also known by the decoder is a maximum offset the start location can be, which consists of the first pixel which cannot belong to any of the previous  $n$  contours. This pixel must then belong to an as yet undefined contour as all pixels in the image are in a contour.

---

<sup>1</sup> This specific choice for describing a contour is possibly not the best way and is not justified in this paper.

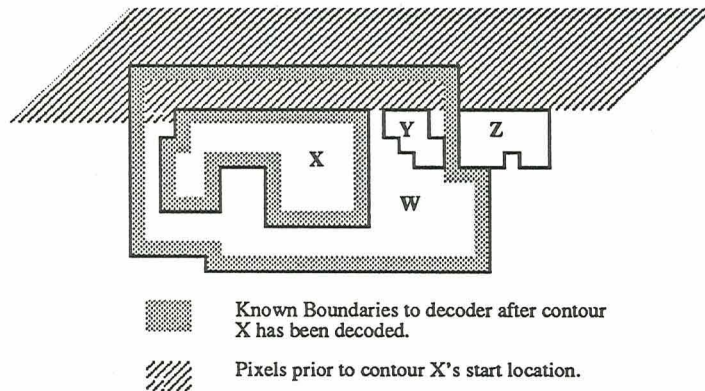


Figure 1. Contour decoding just before contour Y is to be decoded.

The resulting stream has been observed to be highly skewed and to exploit this high probability content it is passed through an Adaptive One Level Conditional Arithmetic Coder.<sup>2</sup>

**2.2.2 Height Values.** Intensity values in an original raster image often have an inherent skew exploited by many prediction based image compressors, the most common being the Differential Pulse Code Modulation (DPCM) family of coders. This effect is drastically reduced when only one height value per contour needs to be coded but it is still possible to take advantage of it. Similarly encoding the difference from previous contour height usually reduces the information content, so difference values are passed to an Adaptive One Level Conditional Arithmetic Coder.

**2.2.3 Boundary Description.** The Boundary Description of a contour can be exceedingly complicated but must be represented exactly. A clockwise cyclic route from the start location around the boundary and returning to the original start location gives a simple path. This code is also comma free, by imposing that you have to end up at the start location. Using a simplified 4-way connected version of Freeman's line code [3], two bits per movement step describe this route fairly compactly, see Figure 2.

On a boundary description there will be as many "north" moves as "south" moves and as many "east" moves as "west" moves. As a boundary description involves a clockwise walk if the relative moves are described in terms of "forward", "backward", "right" and "left" there should be more "right" moves than "left" moves and more "forward" moves than "backward" moves, which can be exploited, so this later format is adopted.

There are two subtly different ways of going around the boundary, either through the centre of pixels or around the edge of pixels. Both of the methods have their own tradeoffs:

#### Movements Through Centres (MTC):

- All four directions are possible, so a maximum of two bits per movement step are required.

<sup>2</sup> An Arithmetic Coder is a probability coder which works on similar principles to the Huffman Coder except with greater efficiency. When coding a symbol a simple table of probabilities for all symbols has to be present for both coder and decoder. The resulting coded size for any particular symbol is smaller the larger its probability of occurrence. The basic design of the Adaptive Arithmetic Coder implementation by Bell [1] was used, changed slightly by adding one level of conditionality, which allows it to deal with conditional probabilities.

- A single pixel contour has to be treated as a special case and a simple system is to code it as a "north" move, which cannot occur as all pixels prior to the start location are represented in previous contours.
- Certain boundaries cannot be represented and have to be represented as two contours. These occur when the cyclic boundary path has to return through the start location before the end. This is demonstrated in Figure 3.

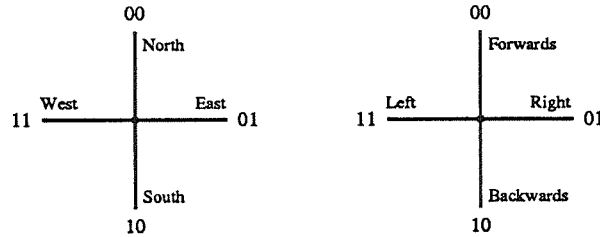


Figure 2. Simplified 4-way Freeman Direction Code, with two alternative labellings.

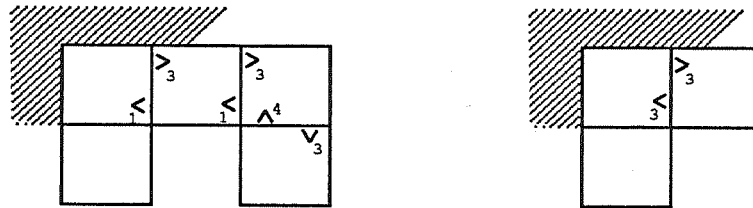


Figure 3. Examples of two simple contours which have to be coded as two contours under MTC. The start locations are in the top left and the numbers indicate the number of choices at each movement stage.

### Movements Along Edges (MAE):

- As you never return along the edge you have just come from only three of the possible directions are valid at any point so a maximum of  $\log_2 3 \approx 1.585$  bits per movement step are required.
- The difference between the number of movement steps varies greatly with MTC. Boundary descriptions increase in size greatly for small contours; 61% increase for a contour of two pixels, and reduce in size for large boundaries.
- All possible contours can be specified and the single pixel contour does not need to be treated as a special case. Although the single pixel now has to be represented by two<sup>3</sup> movement steps.

<sup>3</sup> Note not four steps as all pixels prior to the start location are known meaning that the two movement steps from top left to top right and from bottom left to top left have to be made so are not coded.

Both of these boundaries paths can have their possible movement choices restricted by six external factors:

1. Pixels prior to the start location are all known not to be included in this contour.
2. Horizontal and vertical edges of the full raster image are known.
3. All the contours previously coded will be known by the decoder at this stage and any whose boundary abuts the boundary for this contour will impose movement restrictions.
4. Whilst describing the boundary description this perimeter line can not cross itself so the previous part boundary so far described also acts as a movement restriction.
5. As a clockwise route is taken to define the boundary description pixels on the outside of the part boundary so far described cannot subsequently form part of the remaining section of the boundary.
6. If during the clockwise route a neighbouring pixel is known to be on the contour any pixel which requires a further rotation to the right cannot thus be on the next step.

The following graph in Figure 4 compares the number of bits required to represent the 36 rectangular contours, whose sides are less than seven, with restrictions imposed by factors 1, 4, 5 and 6. The identity line  $y = x$  graphically displays when one method gives a higher compression ratio than the other, and the line  $y = x(\log_2 3)/2$  is the asymptotic line approached as contour sizes increase to infinity. Figure 5 shows an example of a contour being coded by the two methods.

The cases where MAE gives a coding advantage over MTC, are those where a high level of compression already exists. This means that when an image which gives a high compression ratio with MTC will in general give a higher compression ratio with MAE, but when there is a poor compression ratio with MTC there is a resulting poorer compression ratio with MAE. MTC gives an exceedingly low expansion of random data. To create a universal coder which works well on a wide range of images MTC was used.

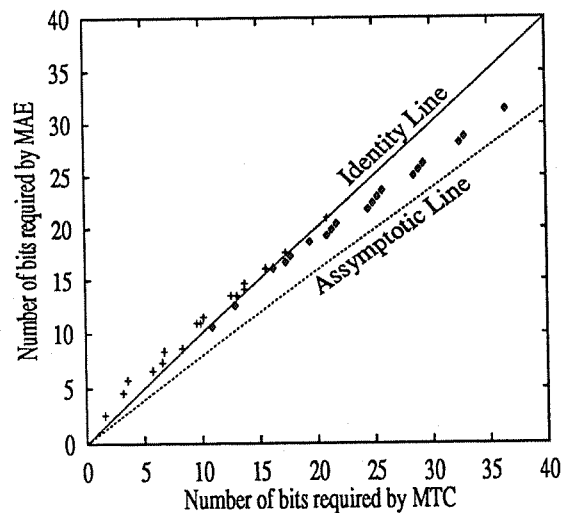
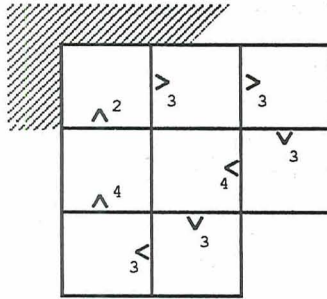


Figure 4. Plot comparing compactness of MTC and MAE for rectangular contours.



### Movement Through Centres

Start at top left pixel heading North.



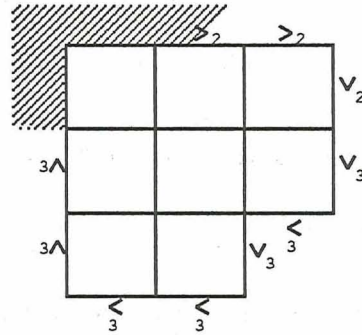
Path : R, F, R, R, L, R, R, F.

Choices: 3, 3, 3, 4, 3, 3, 4, 2.

# bits:  $\log(3)+\log(3)+\log(3)+\log(4)+\log(3)+\log(3)+\log(4)+\log(2) = 12.925$ .

### Movement Along Edges

Start at top right of top left pixel heading East.



Path : F, F, R, F, R, L, R, F, R, F.

Choices: 2, 2, 2, 3, 3, 3, 3, 3, 3, 3.

# bits:  $\log(2)+\log(2)+\log(2)+\log(3)+\log(3)+\log(3)+\log(3)+\log(3)+\log(3)+\log(3) = 14.095$ .

Path gives a list of relative movement steps from "forwards", "backwards", "right" and "left".  
Choices gives the total number of different movement directions known to the decoder.  
# bits calculates the total number of bits required to code this path.

Figure 5. Example of the coding of a simple eight pixel contour using the two alternative methods.

An ideal solution would be to merge the two coding schemes but any simple bookkeeping system either is inefficient or causes too much overhead. Also any merging system would reduce the efficiency of any subsequent probabilistic coding of the stream.

The stream of movement steps is still a very inefficient method of representing a boundary as many movement patterns constitute invalid boundary descriptions.

To reduce this inefficiency a two stage process was designed:

**The first stage** uses as much of the information known by the decoder to restrict the allowable movement directions as listed above. A *Direction Mask* is calculated which flags allowable movement directions.

**The second stage** passes the movement steps through an Arithmetic Coder with a Dynamic Markov Model whose initial model has been set up to detect all impossible movement paths up to a length of eight. This modelling system for the Arithmetic Coder is described in Horspool [6] and allows a fast adaptation of movement paths. Common paths are given a higher probability over further impossible or uncommon paths. The mask from stage one allows current impossible movement directions to be discounted.

## 2.3 Storage

The final image is represented as three separate files or streams and a small header. All of these files can be concatenated together to minimise storage overheads or kept as separate files which incurs only a slight overhead.

## 3. Contour Trees -- A Structured Approach

The list of contours can be very large and at present no ordering system has been imposed. For coding, to aid compaction, a simple first seen order in a raster scan is very useful but for image manipulation it is limiting. A random pixel access operation is as costly as an ordered list search.

To combat this a hierarchical structure called a contour tree was designed which stores those contours totally enclosed by another contour as its children. The structure commences with an imaginary contour defining the image boundary and having all image contours as its children. Siblings can be arranged in some order, first seen in a raster scan is a simple choice. This makes random pixel access times equivalent to a hierarchical order list search. The structure is similar in some senses to a quadtree but its structure is dictated by the original image rather than by arbitrary division.

For image manipulation operations each contour tree leaf can be allocated certain properties, for example; size, bounding boxes and previous historical operations carried out, have proved useful.

## 4. Contour Merging to Aid Lossy Compression

Given a contour tree, for each contour we can assign a rating of noticeability according to its size, actual intensity and neighbourhood. It can then be decided if two contours when merged together, creating a joint contour of equal mean intensity, will be noticed by a human observer up to some fidelity criteria. This gives us a way of calculating a merging threshold value.

By repeated merging until all contours cannot merge any further without going over this fidelity criteria, the resulting contour tree is pruned. Each node is usually more complicated but contour coding should result in a higher compression ratio.

This is akin to work carried out by Samet [9] on quadtrees where under certain conditions the quadtree is repeatedly pruned giving a simplified tree. The concept is very similar except that the choice of pruning is gained from the actual structure of the image rather than the local pixel neighbourhood.

The non-deterministic physiological problem is deciding what criteria constitutes the level of noticeability for the merging of two contours.

The current criteria follows these rules:<sup>4</sup>

1. The size of the larger contour is irrelevant as the eye concerns itself with local changes. It has been observed that details attract attention.

---

<sup>4</sup> General rules regarding the physiological and psychological aspects of the eye were extracted mainly from Boff [2], with countless exceptions from other sources mainly the Physiology Department, Cambridge.

2. The size of the smaller contour is critical and its area is linear with the possibility of being recognised as a separate area. Experiments have shown that noticeability of an object in the human eye is proportional to its area.
3. The intensity difference over background intensity ratio is treated linearly with the possibility of merger. It was discovered that average light intensity seems to follow Weber's Rule.

$$\frac{\delta I}{I} = K \text{ a constant}$$

This means that there exists an  $n$  under certain conditions such that a change in intensity is only noticed when the change in intensity is greater than  $n\%$ . Experimentally this has been shown to be accurate only for average light intensities and short exposure times.

4. The maximum "historic" difference of contours intensity values is considered, not just the current value for each contour. This means as a contour expands its constituent contours' intensity are still considered.
5. The number of times either of the contours have already merged before being considered for this new merge. This affects the threshold value by a linear step. This rule was introduced to compensate for effects caused by the iterative way the merging algorithm is implemented.

A single threshold value is specified and two contours are merged together when the deciding algorithm returns a smaller number.

The resulting image can then be passed to the previously described contour coder.

## 5. Operational Costs

The two operations of contour merging and contour coding are quite separate. Contour merging operations purely pre-filter the image before being transmitted or stored, as shown in Figure 6. The critical time operations then are the contour encoding and decoding.

The costs for contour coding are linear with the size of the image with the decoder requiring less than four random data access per image pixel. The three streams can be coded and transmitted separately splitting the work required to three linear probability coders. The coding process as it presently works requires frame store for the entire picture to be held both at the encoder and decoder which with present systems is not an unacceptable amount of storage.

The costs for the contour merging are more severe being  $O(n<l>)$ , where  $n$  is the number of contours and  $<l>$  is the average length of the resulting contour boundary descriptions.

## 6. Results

For many computer generated images, even rendered and ray-traced images, the resulting compression ratio of the original is reasonably high. The main thorn in the side is with real images, often captured with cameras or scanners, which introduce their own biases and noise. It is these images, whose contour tree description are very flat with an enormous number of small contours, that will be studied in this section.



## 6.1 Test Images

Four non-synthetic images are presented, all are stored at 8bpp gray scale:

- balloons** a famous test image picture, showing a girl within a cascade of smoothly shaded balloons.
- mandrill** the famous test image picture of a monkey's face.
- escher** a 300dpi scanned on an Apple Scanner, sketch drawing by M.C. Escher.
- garden** an image captured by a video camera.

These pictures are shown in Figure 7. Height values in the contour description are mapped directly from the actual gray scale intensity values.

## 6.2 Analysis

The size of the contour trees for each image with the number of contours at each level of the contour tree are shown in the second table. These images have very little structure, the "mandrill" contour tree consists of over 94% single pixel contours. The first table and graph compares compression ratios of the contour coder (CC) with an Adaptive One Level Conditional Arithmetic Coder (AC) as described in Witten [13], the LZT variant in the Lempel-Zif family as first presented in Tischer [10] and a Quadtree Coder (QTC) described in Turner [11]. Also shown is the theoretical compression ratio from entropy calculations using Shannon's formula:

$$E = - \sum_{n=0}^{N-1} p_n \log p_n$$

where  $p_n$  is the probability of pixel value  $n$ , in the range  $0 \dots N-1$ , of occurring.

Second order and third order entropy range over pairs and triples of pixel values, and Diffs is first order entropy over pixel value differences.

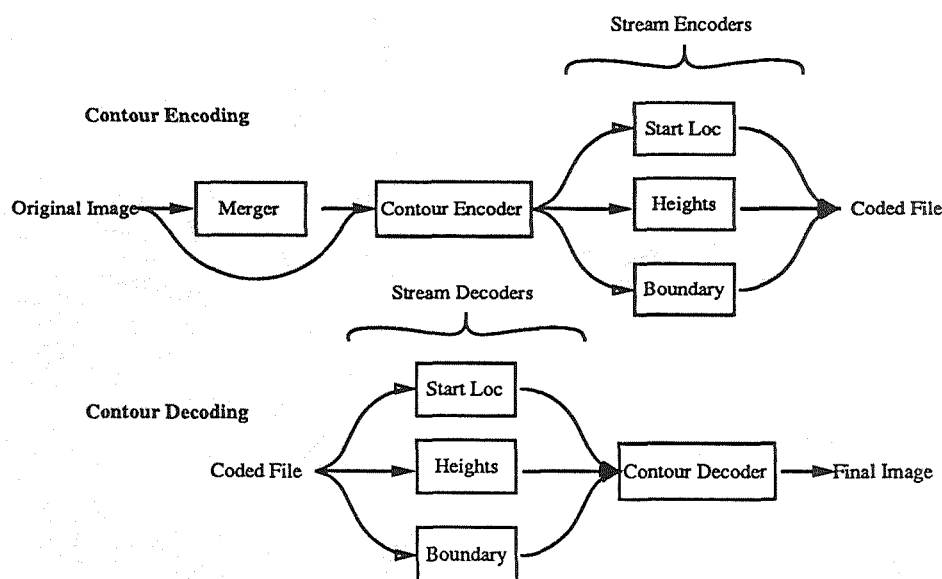


Figure 6. Schematic diagram of the contour coding with and without application of the premanipulating merging algorithm.

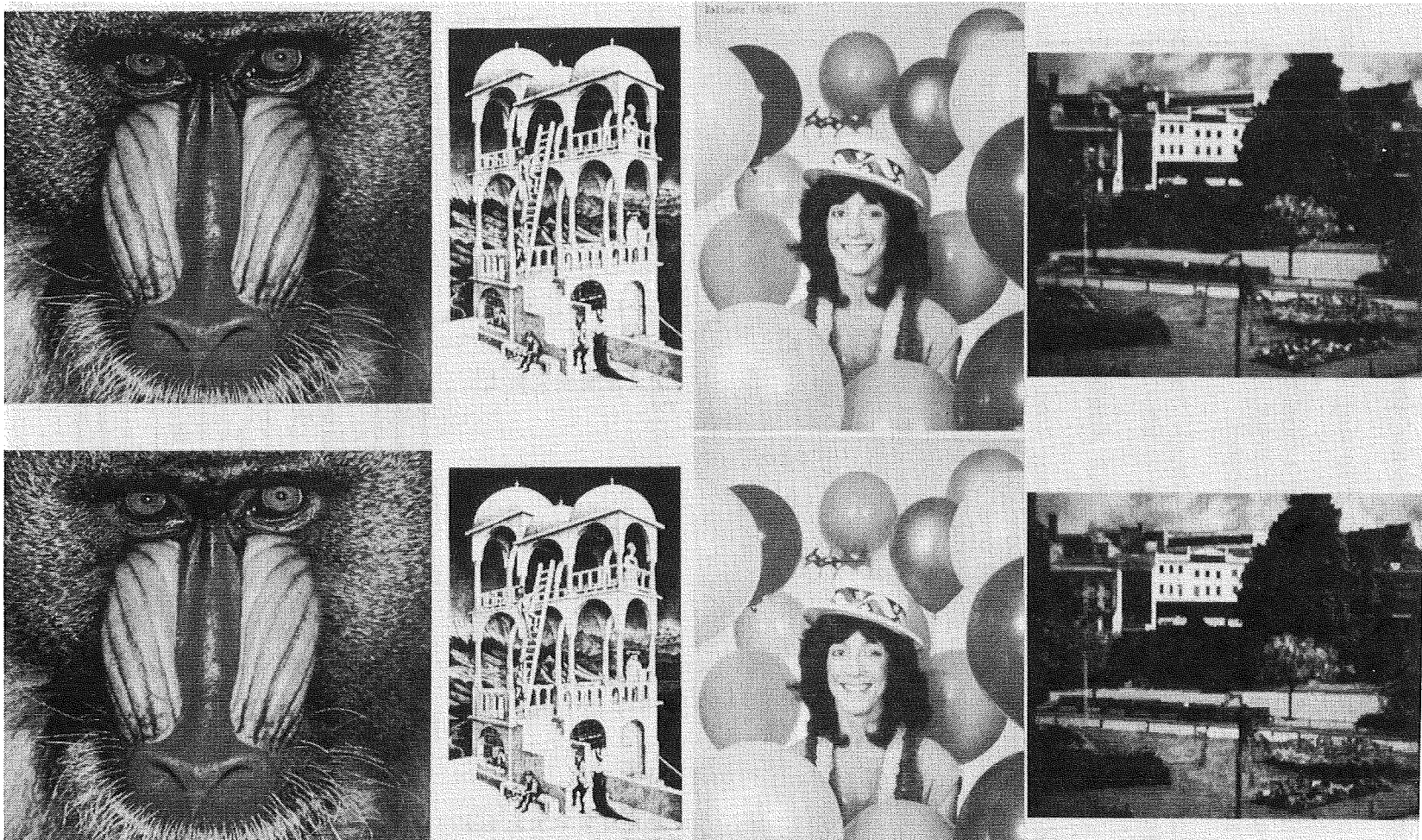


Figure 7. Comparison of originals with *excellent* version: normal colour map.

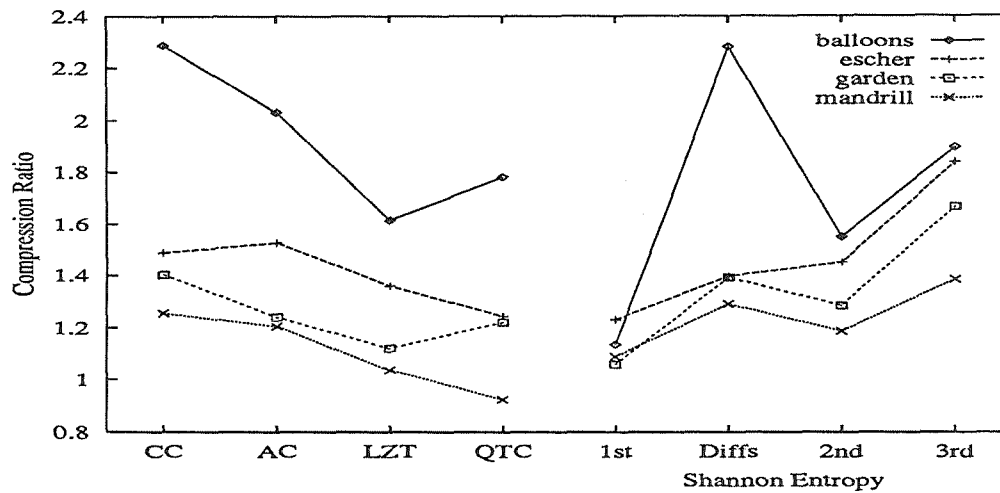


Image	CC	AC	LZT	QTC	1st Order Entropy	Diffs	2nd Order Entropy	3rd Order Entropy
balloons	2.286	2.030	1.614	1.781	1.136	2.280	1.550	1.896
escher	1.488	1.527	1.358	1.244	1.231	1.396	1.450	1.839
garden	1.402	1.240	1.121	1.221	1.060	1.391	1.284	1.667
mandrill	1.254	1.205	1.037	0.924	1.090	1.290	1.187	1.384

To define quality of an image simplified by the contour merging algorithm the term *excellent* was devised which has the simple criteria that a subjective observer cannot tell the original from the merged version when shown separately. These images are shown in Figures 7 & 8. Compression results achieved with this "excellent" criteria. The next table quantifies the number of contours and compression ratios achieved, in the original (first line) and "excellent" images (second line).

Image	Size:pxls	Comp. Ratio	No. Contours	Hierarchical Depth		
balloons	198664	2.286	85930	84652	1278	
		5.771	5892	4925	966	1
escher	138632	1.488	91546	1	91199	346
		4.715	8443	1	7299	1143
garden	49160	1.402	40613	40498	116	
		4.356	3107	2887	219	1
mandrill	245768	1.254	231086	231086		
		3.930	19744	17823	1921	

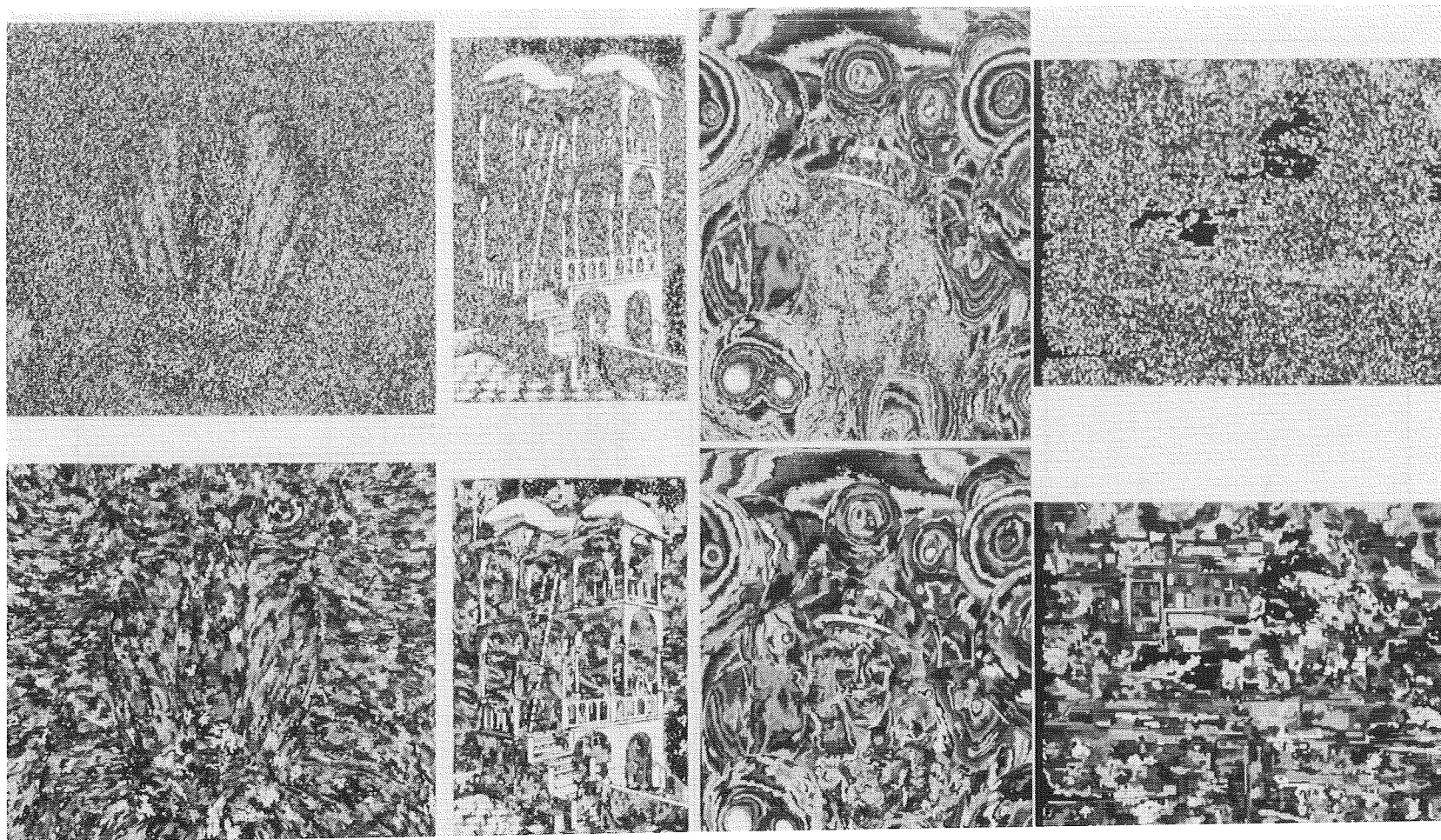


Figure 8. Comparison of originals with *excellent* versions: random colour map.



### 6.3 Visual Artifacts

Those images which are "excellent" can still have up to one fifteenth the number of contours which with the contourcoder gives compression ratios between 4:1 and 6:1. When larger threshold values are given to the contour merging algorithm artifacts start to be seen. The main artifacts are:

**Loss of Highlights** as in the flattening of the pupil of the mandrill's eye, shown in Figure 9.

**Loss of Connectivity** between regions which the mind psychologically fills in. This can be seen in the top railing on the image "escher" in Figure 10. Merging operations on the left and right of the already partially disconnected railing separates it even further.

**Blotching** occurs due to a high level of merging. Noticeable in areas of randomness an example is in the flower bed of the image "garden", as shown in Figure 11.

**False Contouring** occurs when smoothly shaded areas group into regions. Noticeable on low quality printers which add to the effect, due to intensity quantization.

Two features which the merging algorithm exploits with its simple set of rules:

**Edges are Sharpened** up which according to Marr [7] are particularly recognisable in the human brain.

**Texture Shape** via the contour merging algorithm is detected at the low level and the merging process flows with it. This allows a way of simplifying and describing areas such as grass and hair.

## 7. Conclusions

Demonstrated was a new technique in lossy image compression which offers an alternative to other currently used methods. When combined with an up to date contour coder the initial results are very promising and demonstrate that its use is viable.

## References

- [1] T. C. Bell, J. G. Cleary, I. H. Witten, *Text Compression*, Prentice Hall Advanced Reference Series.
- [2] K. R. Boff and J. E. Lincoln (ed), *Engineering Data Compendium, Human Perception and Performance*, Harry G. Armstrong Aerospace Medical Research Lab.
- [3] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," *IRE Transactions on Electronic Computers*, June 1961.
- [4] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Second Edition, Addison-Wesley Pub. Advanced Book Program.
- [5] D. N. Graham, "Image Transmission by Two-Dimensional Contour Coding," *Proceedings of the IEEE*, Vol. 55, No. 3, March 1967.
- [6] R. N. Horspool and G. V. Cormack, "Dynamic Markov Modelling -- A Prediction Technique," *Proc. International Conference on the System Sciences*, Honolulu, HA, January 1986.
- [7] D. Marr, *Vision*, Freeman.
- [8] W. Pratt, *Digital Image Processing*. Second Edition. Wiley-Interscience, ISBN 0-471-85766-1.

- [9] H. Samet, "Data Structures for Quadtree Approximation and Compression," *Comms. of the ACM.*, Vol. 28, No. 9, Sept. 1985.
- [10] P. Tischer, "A Modified Lempel--Ziv-Welch Data Compression Scheme," *Australian Computer Science Communications*, Vol. 9, No. 1, pp. 262--272.
- [11] M. J. Turner, "ICEBERG - Gray Scale Image Compression Scheme Part II," *Computer Science Tripos 1990*, Dissertation, Cambridge University, UK.
- [12] T. A. Welch, "A Technique for High Performance Data Compression," *IEEE Computing Society, Computing*, Vol. 17, No. 6, June 1984.
- [13] I. H. Witten, R. Neil, J. G. Cleary, "Arithmetic Coding for Data Compression," *Comms. of the Assoc. for Comp. Mac.*, Vol. 30, No. 6, June, pp. 520--540.

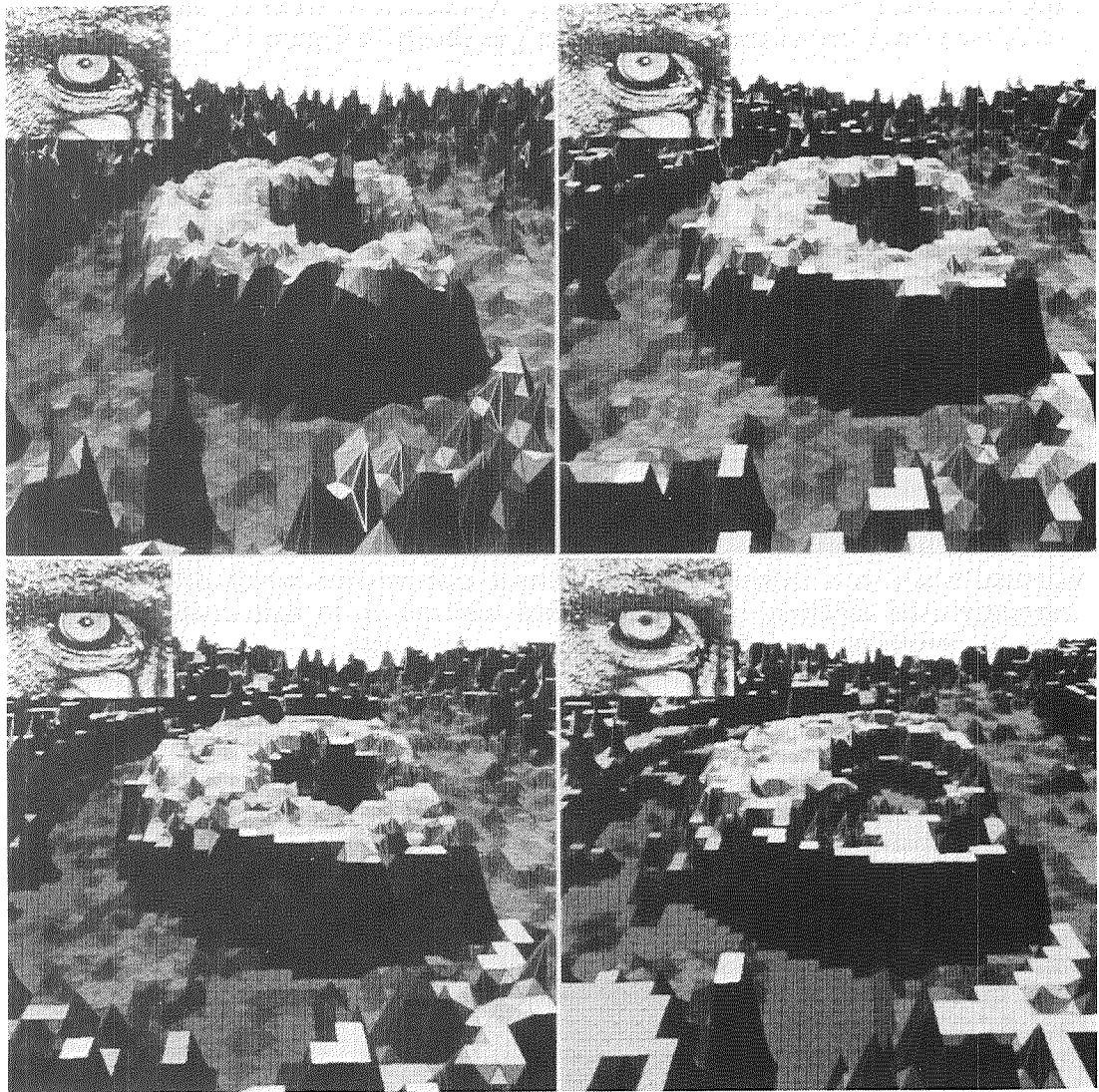


Figure 9. Shown is a sequence of 3D representations of an image which demonstrates how increased threshold values affect the Contour Merging process. The 3D view shows the contours around the mandrill's right eye as seen when standing above the bridge of the nose.



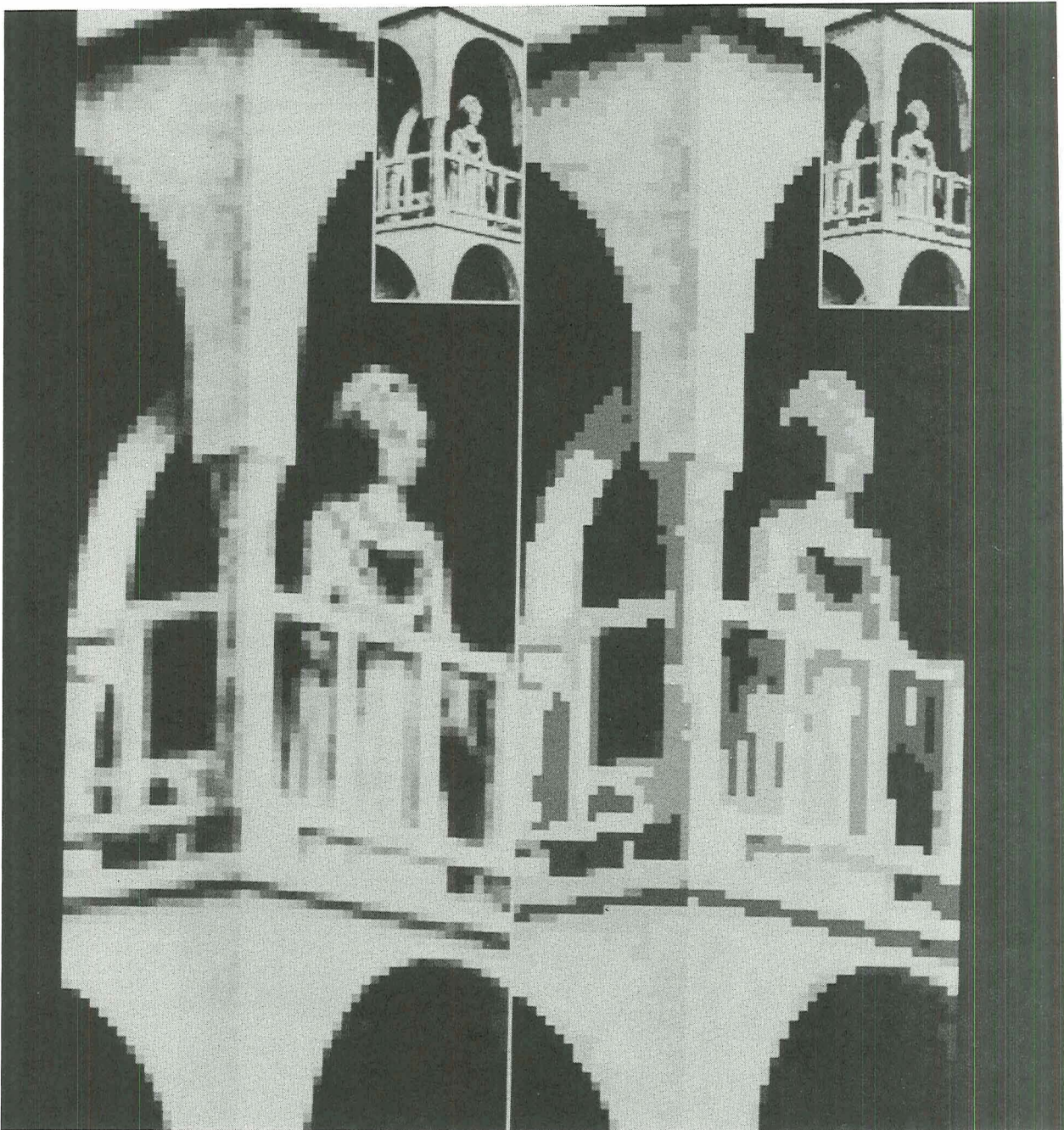


Figure 10. Shown is a blown-up section of the image "eshcer." The image on the right has been coded using the contour merging algorithm. Visual artifacts includes loss of connectivity, slight false contouring, and loss of highlight.

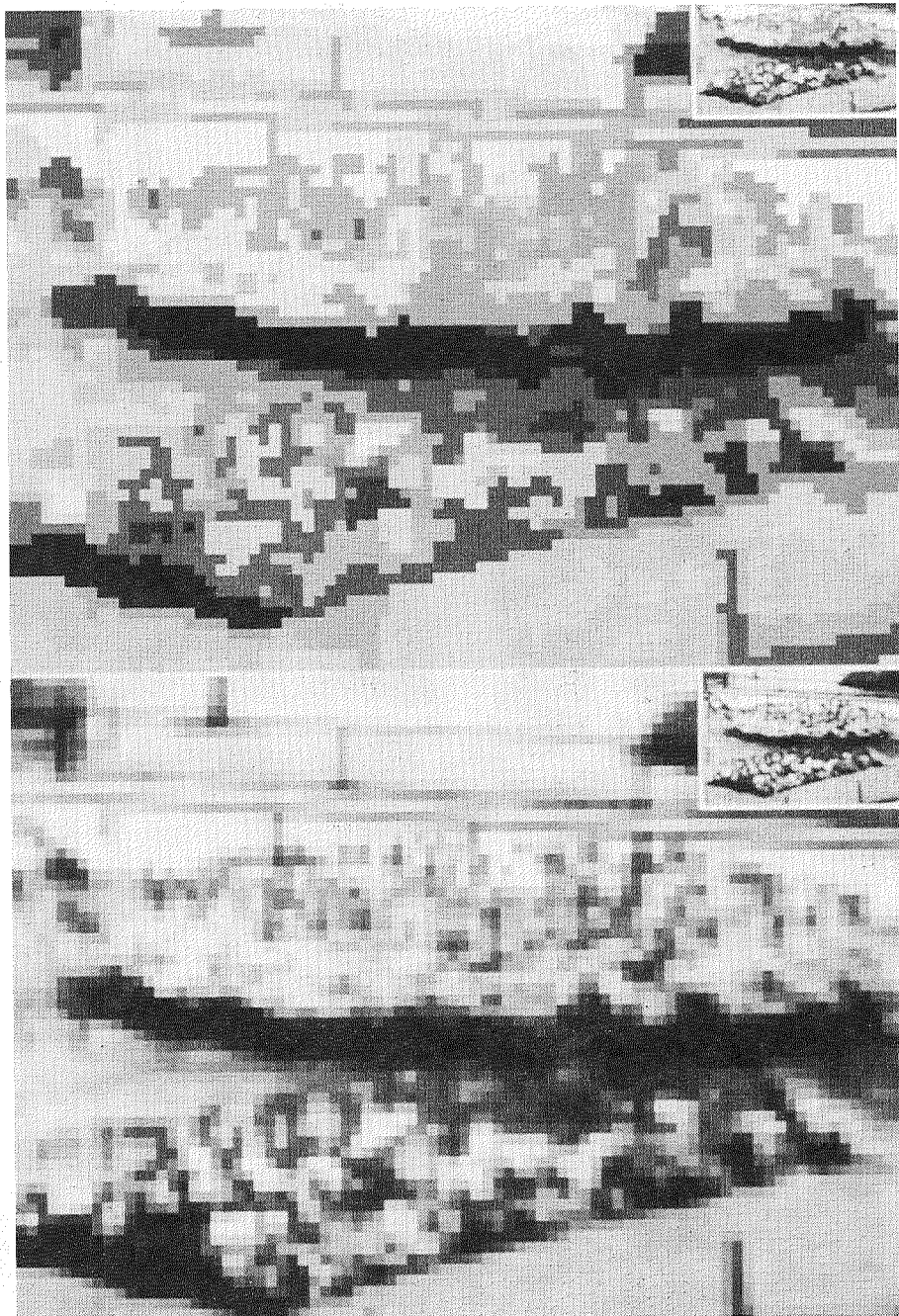


Figure 11. Shown is a blown-up section of the image "garden." The flower bed becomes blotchy large threshold value.